AD-A265 806

ENTATION PAGE

age 1 hour per response, including the time for reviewing instruction , searching existing data sources, gathering and
formation Send comments regarding this burden estimate or any other aspect of this collection of information including
es, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204 Arlington, VA
93 _____ and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503

1 AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE March 1993	3 REPORT TYPE AND DATES COVERED Professional paper

4. TITLE AND SUBTITLE ADVANCED DISPLAY OBJECT SELECTION METHODS FOR ENHANCING USER-COMPUTER PRODUCTIVITY	5 FUNDING NUMBERS PR: CDB8 PE: 0602234N WU: DN309119

6 AUTHOR(S)

G. A. Osga

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Command, Control and Ocean Surveillance Center (NCCOSC) RDT&E Division San Diego, CA 92152–5001	8 PERFORMING ORGANIZATION REPORT NUMBER

9 SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Office of Naval Technology Office of Chief of Naval Research Arlington, VA 22217	10 SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b DISTRIBUTION CODE

DTIC
ELECTE
JUN 17 1993
S
A
D

13. ABSTRACT *(Maximum 200 words)*

The User-Interface Technology Branch at NCCOSC RDT&E Division has been conducting a series of studies to address the suitability of commercial off-the-shelf (COTS) graphic user-interface (GUI) methods for efficiency and performance in critical naval combat systems. This paper presents an advanced selection algorithm and method developed to increase user performance when making selections on tactical displays. The method has also been applied with considerable success to a variety of cursor and pointing tasks. Typical GUIs allow user selection by: 1) moving a cursor with a pointing device such as a mouse, trackball, joystick, touchscreen, and 2) placing the cursor on the object. Examples of GUI objects are the buttons, icons, folders, scroll bars, etc. used in many personal computer and workstation applications. This paper presents an improved method of selection and the theoretical basis for the significant performance gains achieved with various input devices tested. The method is applicable to all GUI styles and displays sizes, and is particularly useful for selections on small screens such as notebook computers. Considering the amount of work-hours spent pointing and clicking across all styles of available graphic user-interfaces, the cost/benefit in applying this method to graphic user-interfaces is substantial, with the potential for increasing productivity across thousands of users and applications.

93-13468

14 SUBJECT TERMS man-machine interface and ASW decision support operator functions	decision making controls and displays decision aids	15 NUMBER OF PAGES
		16 PRICE CODE

17 SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18 SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19 SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20 LIMITATION OF ABSTRACT SAME AS REPORT

21a. NAME OF RESPONSIBLE INDIVIDUAL	21b. TELEPHONE (include Area Code)	21c. OFFICE SYMBOL
G. A. Osga	(619) 553–3644	Code 441

Accession For

NTIS GRA&I	✓
DTIC TAB	☐
Unannounced	☐
Justification	

By
Distribution /

Availability Codes

Dist	Avail and/or Special
A-1	

DTIC QUALITY INSPECTED 3

ADVANCED DISPLAY OBJECT SELECTION METHODS FOR ENHANCING USER-COMPUTER PRODUCTIVITY

Dr. Glenn A. Osga
Naval Command Control & Ocean Surveillance Center,
RDT&E Division
San Diego, Ca. 92152-5000

ABSTRACT

The User-Interface Technology Branch at NCCOSC RDT&E Division has been conducting a series of studies to address the suitability of commercial off-the-shelf (COTS) graphic user-interface (GUI) methods for efficiency and performance in critical naval combat systems. This paper presents an advanced selection algorithm and method developed to increase user performance when making selections on tactical displays. The method has also been applied with considerable success to a variety of cursor and pointing tasks. Typical GUIs allow user selection by: 1) moving a cursor with a pointing device such as a mouse, trackball, joystick, touchscreen, and 2) placing the cursor on the object. Examples of GUI objects are the buttons, icons, folders, scroll bars, etc. used in many personal computer and workstation applications. This paper presents an improved method of selection and the theoretical basis for the significant performance gains achieved with various input devices tested. The method is applicable to all GUI styles and display sizes, and is particularly useful for selections on small screens such as notebook computers. Considering the amount of work-hours spent pointing and clicking across all styles of available graphic user-interfaces, the cost/benefit in applying this method to graphic user-interfaces is substantial, with the potential for increasing productivity across thousands of users and applications.

INTRODUCTION

Many varieties of cursor selection tasks exist across graphic user-interface (GUI) applications. Selectable objects include: text on menus, radio buttons, check boxes, icons, file folders, buttons, and labels. They may also include graphic objects in computer-aided design, architectural, design and graphics layout programs. The selection of objects by cursor placement can be described in human performance terms by Welford's variation of Fitts's Law, (refs. 1-2). This law states that, within certain limits, cursor distance moved to the object and object size affect user performance time. When positioning time is plotted by Fitts's index of difficulty ($\log^2$ Distance/Size + .5) performance speed is affected by design factors such as control/display ratio, cursor velocity, and quality of visual feedback (refs. 2-4). These critical design parameters related to task difficulty (e.g. distance and object size), and quality of visual feedback were manipulated to yield the advanced selection algorithm (ASA) described in this paper.

The first parameter manipulated by the ASA is cursor-object distance. Current user-interface designs typically require the user to place the cursor on the desired object. Thus, distance traveled by the cursor is dependent on the distribution of objects, size of the display, and efficiency of the user-interface design with respect to cursor travel. In contrast, the ASA reduces the cursor travel distance required by determining the distance from the cursor to the closest selectable object, not requiring cursor placement on the object to be selected.

The second parameter is directly related to cursor-object distance calculation. For selection purposes, objects are treated as larger than they visually appear on the display due to the calculation of relative object-cursor distance. Thus, an object's selection area is directly related to the spacing and distribution of objects, vs. their apparent size.

The third parameter provides the user with necessary constant visual feedback to allow variable cursor-object distances. Methods of visual feedback have become "standardized" across many software applications. With the exception of pull-down or pop-up menus, visual feedback (highlighting) is typically presented after the user performs an action selecting an object. The ASA displays visual highlighting for a "selectable" object constantly as the cursor is moved on the display. Thus, object selectability is shown to the user before a selection action is made, and is independent of object size or absolute location.

METHOD

Complementary software methods were developed to implement the Advanced Selection Aid (ASA). The "selectable" cursor target, defined as the object closest to the current cursor position, is constantly determined. Methods used are dependent on the type of task performed. The first method applies to the selection of objects which are spaced or arranged irregularly on the display. As shown in Figure 1, the computer determines x and y coordinates of object and cursor "hotspot" location. The minimum distance from the cursor to the closest object can be computed using the Pythagorean Theorem. If the cursor is on the same x or y axis as the closest object (e.g. either x or y is less than the object height or width) then the absolute value of x or y is the calculated distance. This distance to all displayed objects is computed in real-time as the cursor is moved. If the cursor was equidistant from two or more objects, the last closest object is used. In Figure 1, object 2 is closer to the cursor and the expanded outline around it indicates that it is currently "selectable" and would be selected if an appropriate selection action is taken.
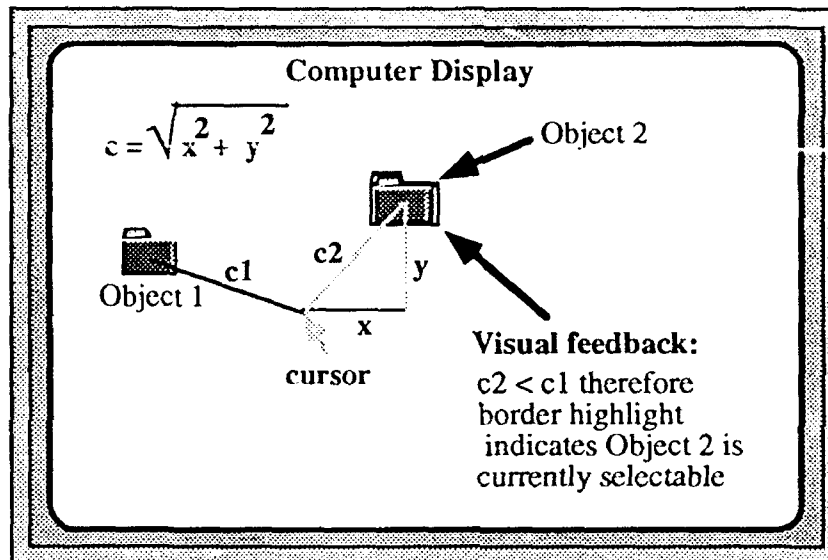

Computer Display

$$c = \sqrt{x^2 + y^2}$$

Object 2

c1

c2

y

Object 1

x

cursor

Visual feedback:

c2 < c1 therefore border highlight indicates Object 2 is currently selectable

Figure 1: Advanced Selection Algorithm cursor-object distance calculation and visual feedback example

For display items which are adjacent and fixed in relative location to each other, such as dialogue window items, the selection area can be determined by defining selection areas which surround each selectable item. A simpler method defines the distance between the cursor and the edge of the object. With collections of irregular shaped objects, the edge must be used to calculate cursor-object distance in lieu of the object center, or results are not accurate. In Figure 2, all objects have larger selection areas than their physical appearance alone would indicate, as illustrated by the selection areas for "Check Box 3" and the "HELP" button. The ASA calculates the distance from the cursor to the nearest object edge, which is "Check Box 3" in this example. The object is highlighted by appropriate methods such as the outlined box shown. Other highlighting methods can be used to indicate the "selectable" object such as color, or outline. We used inverse video and the typical GUI check box or radio button indicators to indicate final object selection.

The ASA impact on machine performance using available PC technology has been negligible. We have tried hundreds of objects simultaneously displayed on a 19" color monitor in 8-bit color mode without any performance decrement. Being computationally simple, the effect of visual highlighting and cursor-object distance computation does not appear to slow other machine calculations. The net effect of these computations and visual highlighting is a closed-loop human-computer control system which does not require the user to make precise cursor movements to accomplish selection tasks. The next section describes the impact of the ASA on human performance.
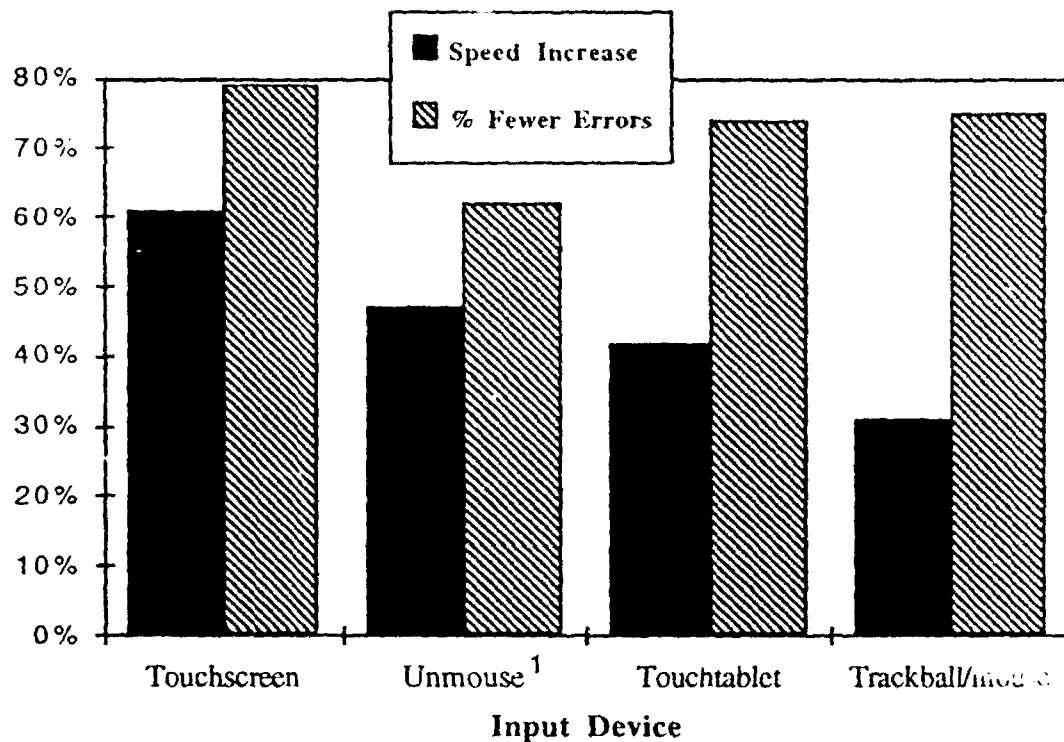
THIS
PAGE
IS
MISSING
IN
ORIGINAL
DOCUMENT

FIG-2

Figure 3: Average performance increase in user speed and accuracy during selection of graphic objects using the advanced selection algorithm[1]

APPLICATIONS

Current GUI users are faced with a myriad of small objects which are constant targets for cursor placement throughout the working day. Software designers include small squares, icons and objects depicting window close boxes, "sizing icons", etc. across all popular GUIs. These selection tasks become more difficult on smaller screens such as portable or notebook computers. The ASA can be applied to computer desktop GUI applications to support tasks such as the manipulation of file folders or window icons. Icons can be displayed with enhanced borders or inverse colors. We have successfully modified dialogue windows in the Apple Macintosh[2] GUI to incorporate the ASA.

Cursor travel distances become larger as two-page size displays are used in workstations and desktop publishing applications. The ASA is particularly useful for these configurations. Touchtablet devices which are designed as small replacements for a mouse or trackball, become difficult to use with large screens due to small finger movements resulting in large cursor movements. The ASA improves performance with small tablets considerably. Several negative design aspects typically associated with touchscreen use are virtually eliminated with the ASA. Cursor offset relative to the finger placement and the impact of obscuring of display objects is reduced. Display objects do not require reformatting with larger buttons or selection areas. Other selection techniques such as eye or head cursor tracking and selection would also benefit from ASA use.

Many GUI programs and applications are impractical for use without pointing and clicking. A segment of the user population with marginal cursor manipulation and pointing skills who are unable to

[1] Unmouse is a trademark of MicroTouch Inc.
[2] Macintosh is a trademark of Apple Computer Inc

comfortably use GUI methods may find point-and-click computing to be practical with ASA implemented. ASA sensitivity can be varied. For example, we are also implementing other selectability criteria to be combined with cursor-object distance, such as the type of object. When selecting from a particular class of objects on a cluttered display containing many object types, a function could be applied which selectively allows the user to apply ASA to specific object classes while excluding others.

CONCLUSIONS

Significant user performance enhancement has been shown in user experiments using ASA for object selection. These effects have been demonstrated for a variety of pointing devices, and should apply to all cursor pointing methods. We have observed no negative effects in machine speed or processing delays in a typical PC configuration. Application of this method is possible in any graphic interface application where cursor-object distance can be computed. Considering the labor-hours spent by thousands of graphic user-interface users moving and placing a cursor for pointing and selection tasks, the application of this technique by software developers will substantially improve productivity across a broad base of end users and applications.

ACKNOWLEDGMENTS

REFERENCES

1. Fitts, P.M. (1954) The information Capacity of the Human Motor System in Controlling Amplitude of Movement, *Journal of Experimental Psychology*, 47, 381-391.

2. Card, S.K, Moran, T.P. and Newell, A. (1983) *The Psychology of Human Computer Interaction*, Lawrence Erlbaum Associates, Hillsdale New Jersey, p 55.

3. Parng, A.K. (1988) *An Automated Test of Fitts' Law and Effects of Target Width and Control-Display Gain Using a Digitizer Tablet*, Technical Document 1214, Naval Ocean Systems Center, San Diego.

4. Greenstein, J.S. & Arnaut, L. Y. (1988) Input Devices in M. Helander, (Ed.), *Handbook of Human-Computer Interaction*, Elsevier, New York.

5. Osga, G.A. (1991) Using Enlarged Target Area and Constant Visual Feedback to Aid Cursor Pointing Tasks *In Proceedings of the Human Factors Society 35th Annual Meeting*, San Fransisco.